

1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details of the selected packet header window” (refer to Figure 2 in the “Getting Started with Wireshark” Lab if you’re uncertain about the Wireshark windows.

Source IP: 210.94.194.77

Port: 1742

2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

Destination IP: 128.119.245.12

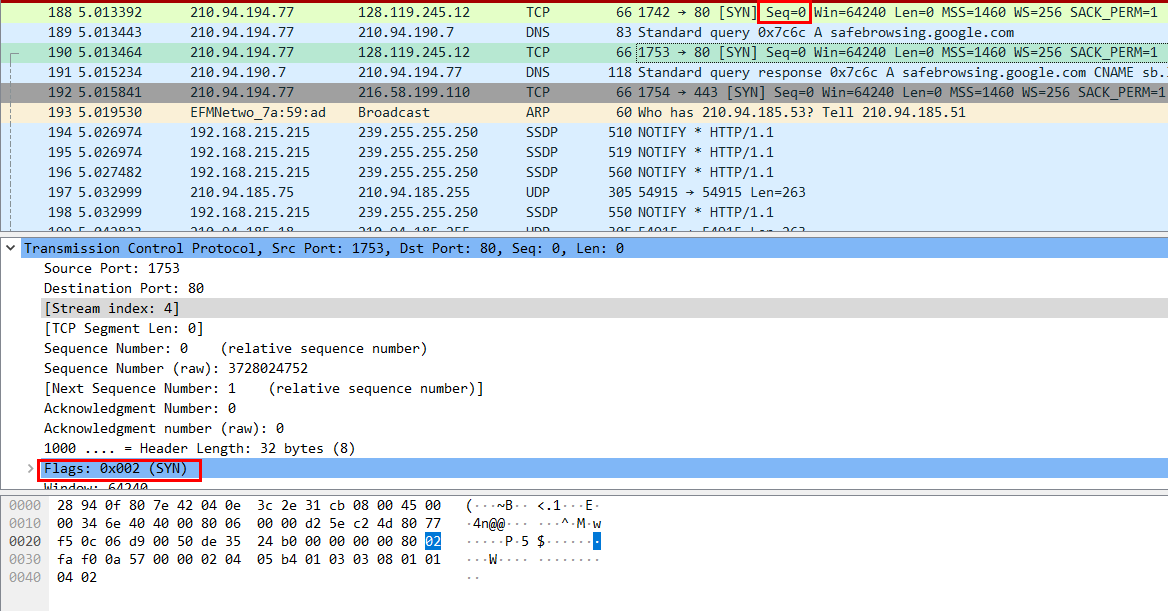
Port: 80

3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

My IP: 210.94.194.77

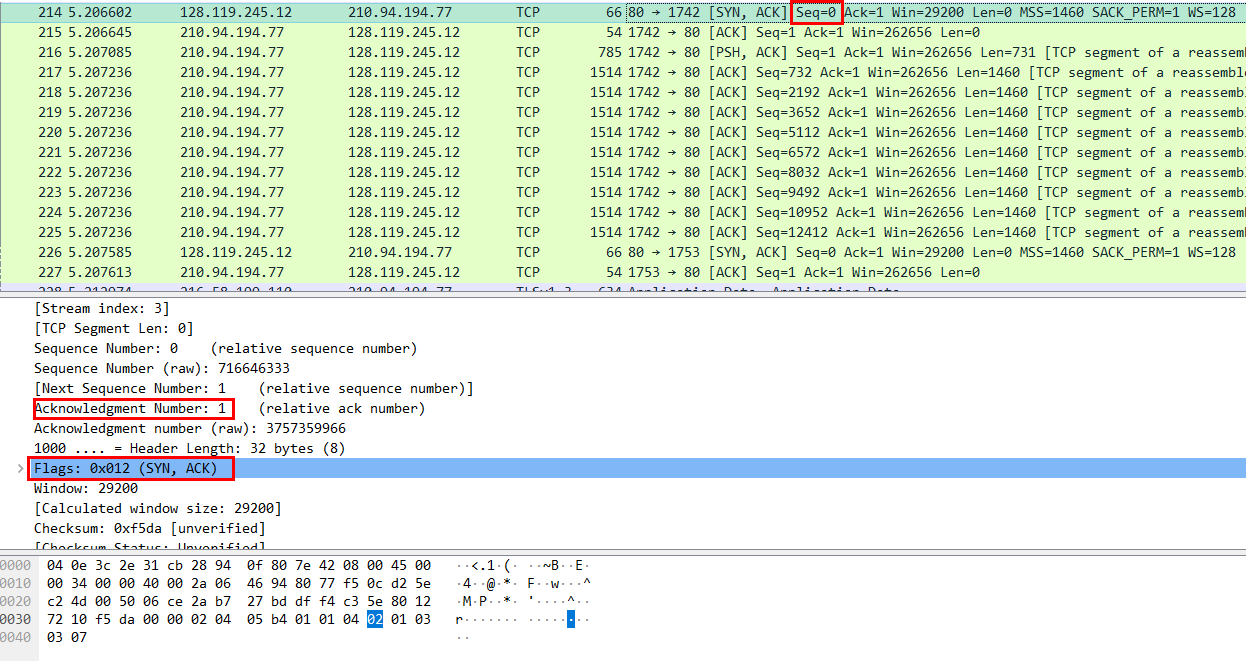
Sending Port: 1742

4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?



TCP connection을 시작하는 segment의 Sequence number는 0이다. SYN플래그가 포함되어있어 이것으로 SYN segment임을 확인할 수 있다

5. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?



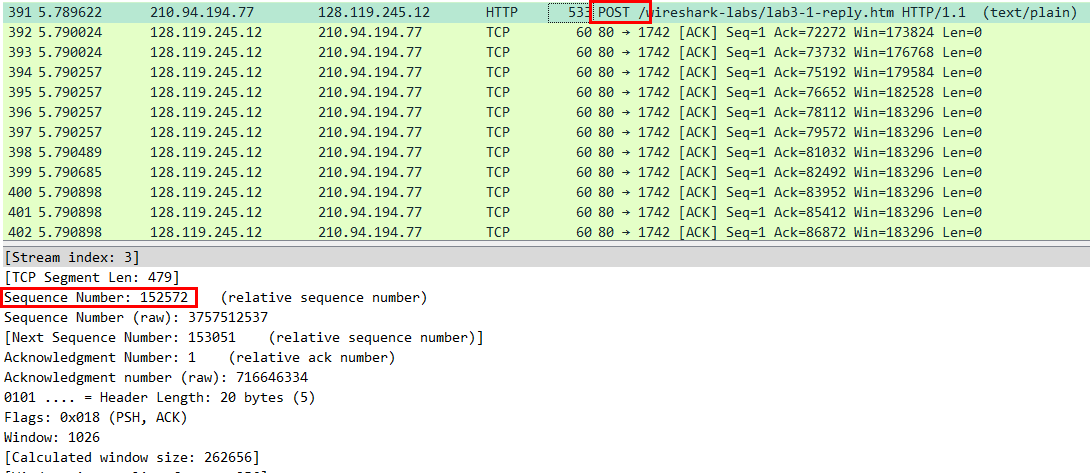
SYNACK 세그먼트의 sequence number는 0이다.

Acknowledgement 필드 값은 1이다.

gaia.cs.umass.edu가 sequence number에서 1을 더함으로써 acknoweldgement 값을 결정한다

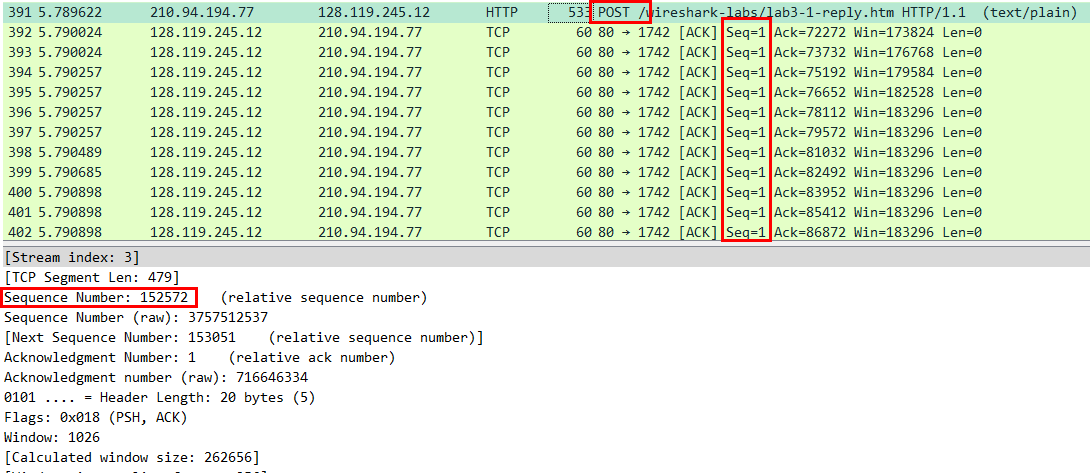
SYNACK 세그먼트에는 SYNACK 플래그로 이것이 SYNACK segment임을 확인한다

6. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you’ll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a “POST” within its DATA field.



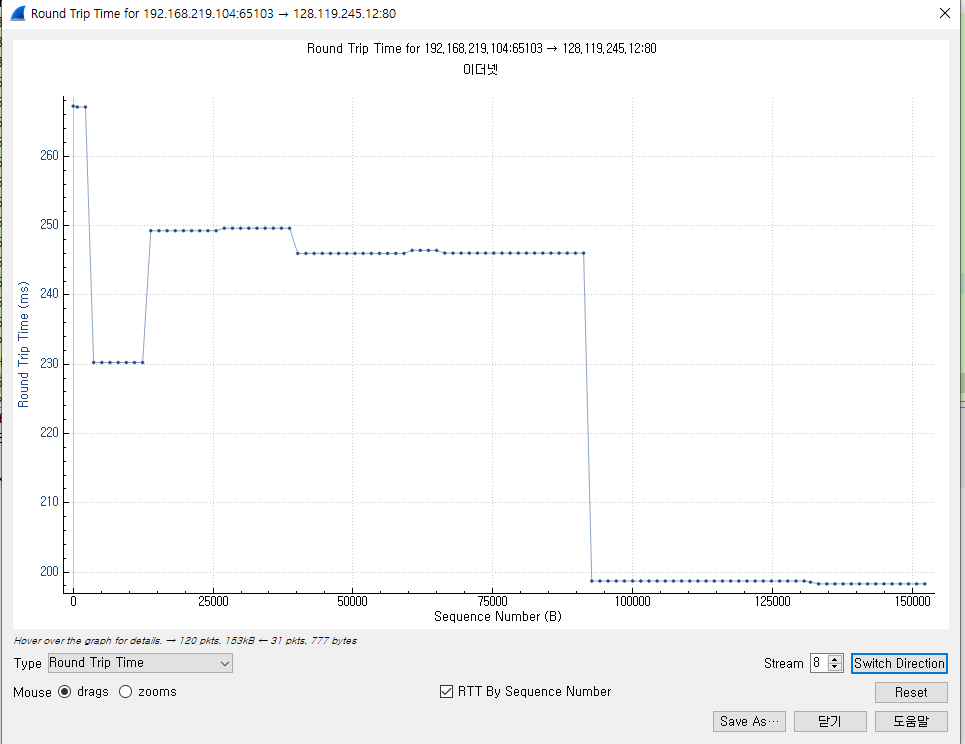
HTTP POST command의 sequence number는 152244이다.

7. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)? At what time was each segment sent? When was the ACK for each segment received? Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments? What is the EstimatedRTT value (see Section 3.5.3, page 239 in text) after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 239 for all subsequent segments.



테이블이(가) 표시된 사진

자동 생성된 설명



위의 사진들을 이용하여 계산한다.

Seq=1, 송신 시각=27.178103, 수신 시각=27.375051, RTT=송신-수신=0.196948

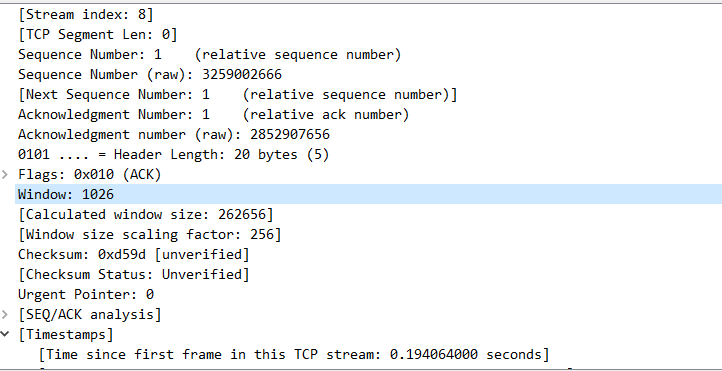
Seq=153053, 송신=27.378715 ,수신=27.38850 ,RTT=0.009785

TCP가 3개까지밖에 안나오는듯합니다

8. What is the length of each of the first six TCP segments?

처음은 863, 다음은 831, 54, 60, 60, 60이다.

9. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

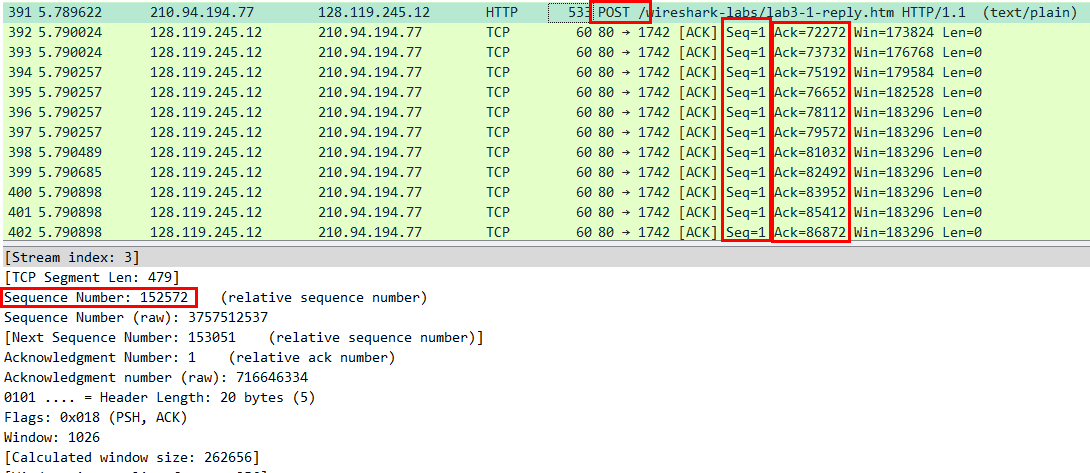


가능한 버퍼 공간은 1026바이트이다.

10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

retransmit가 발생하지 않았다.

11. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 247 in the text).



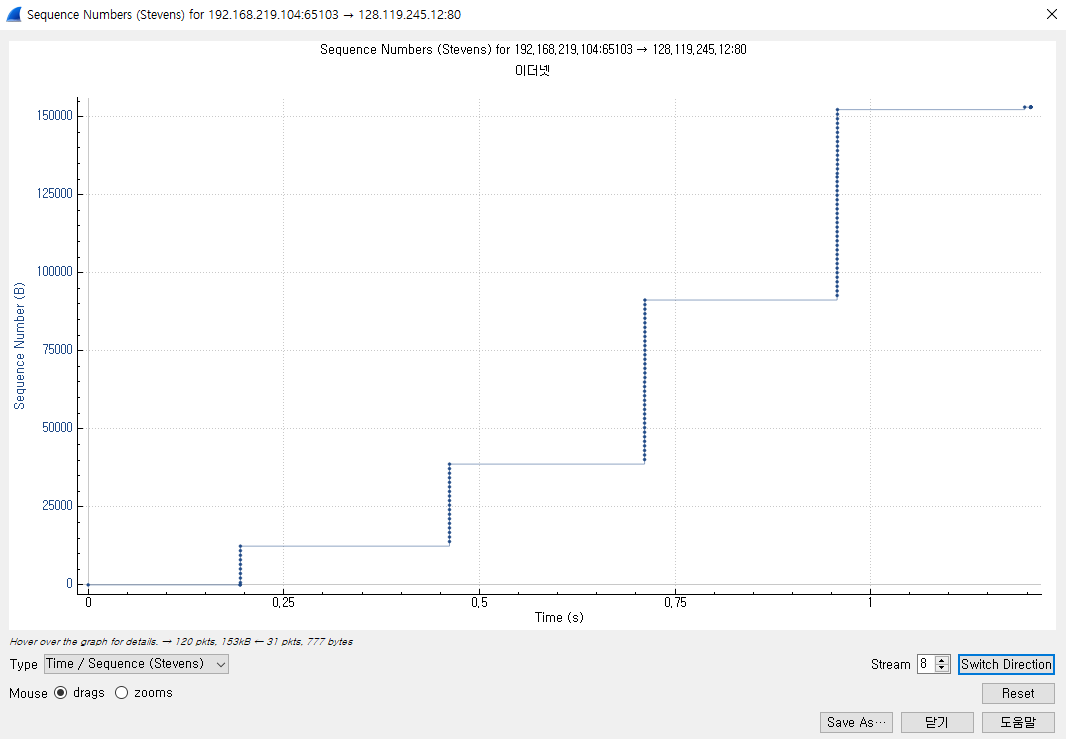
위의 사진의 ACK 숫자를 보면 72272,73732,75192,76652…로 1460씩 증가하는 것을 볼 수 있다. 1460byte가 ACK packet의 크기로 볼 수 있다.

12. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

마지막 ACK와 처음 ACK의 값을 빼고 first frame으로 나눈다. 마지막 ACK의 크기는 153051byte, 처음 ACK의 크기는 1byte이고 first frame은 2.12sec였다

(153051-1)/2.12=72,193.40bps 이다.

13. Use the Time-Sequence-Graph(Stevens) plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP’s slowstart phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we’ve studied in the text.



sequence number가 점점 크게 증가하고 계단식으로 증가하는 것으로 보아 slow start가 지속되고 있다는 것을 알 수 있다.